(12) UK Patent Application (19) GB (11) 2 253 697 (13) A

(43) Date of A publication 16.09.1992

(21) Application No 9202752.3

(22) Date of filing 10.02.1992

(30) Priority data
(31) 9105358 (32) 13.03.1991 (33) GB

(71) Applicant
The Secretary of State for Defence
Defence Research Agency, Meudon Avenue, Farnborough, Hants, GU14 7TU, United Kingdom

(72) Inventor
Stephen Piers Luttrell

(74) Agent and/or Address for Service
Dr A W S Williams
Ministry of Defence, Procurement Executive, Pats 1A4, Room 2014, Empress State Building, Lillie Road, London, SW6 1TR, United Kingdom

(51) INT CL^5
G01N 21/88

(52) UK CL (Edition K)
G1A AAJ AA9 AC1 AC13 AC3 AD1 AG10 AG13 AG6 AR7 AT14 AT26 AT27 AT3 AT4 AT8

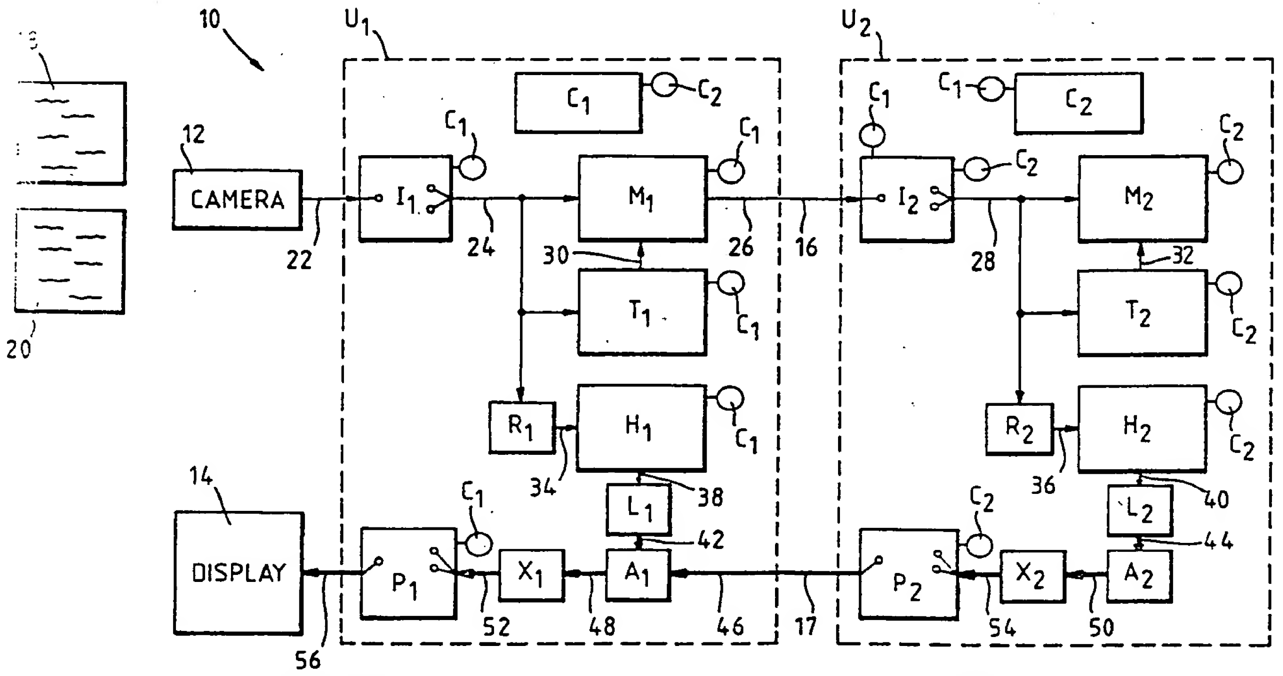
(56) Documents cited
EP 0243639 A2 EP 0048568 A2

(58) Field of search
UK CL (Edition K) G1A AAJ
INT CL^5 G01N 21/88
Online database: W P I

(54) Image anomaly detector

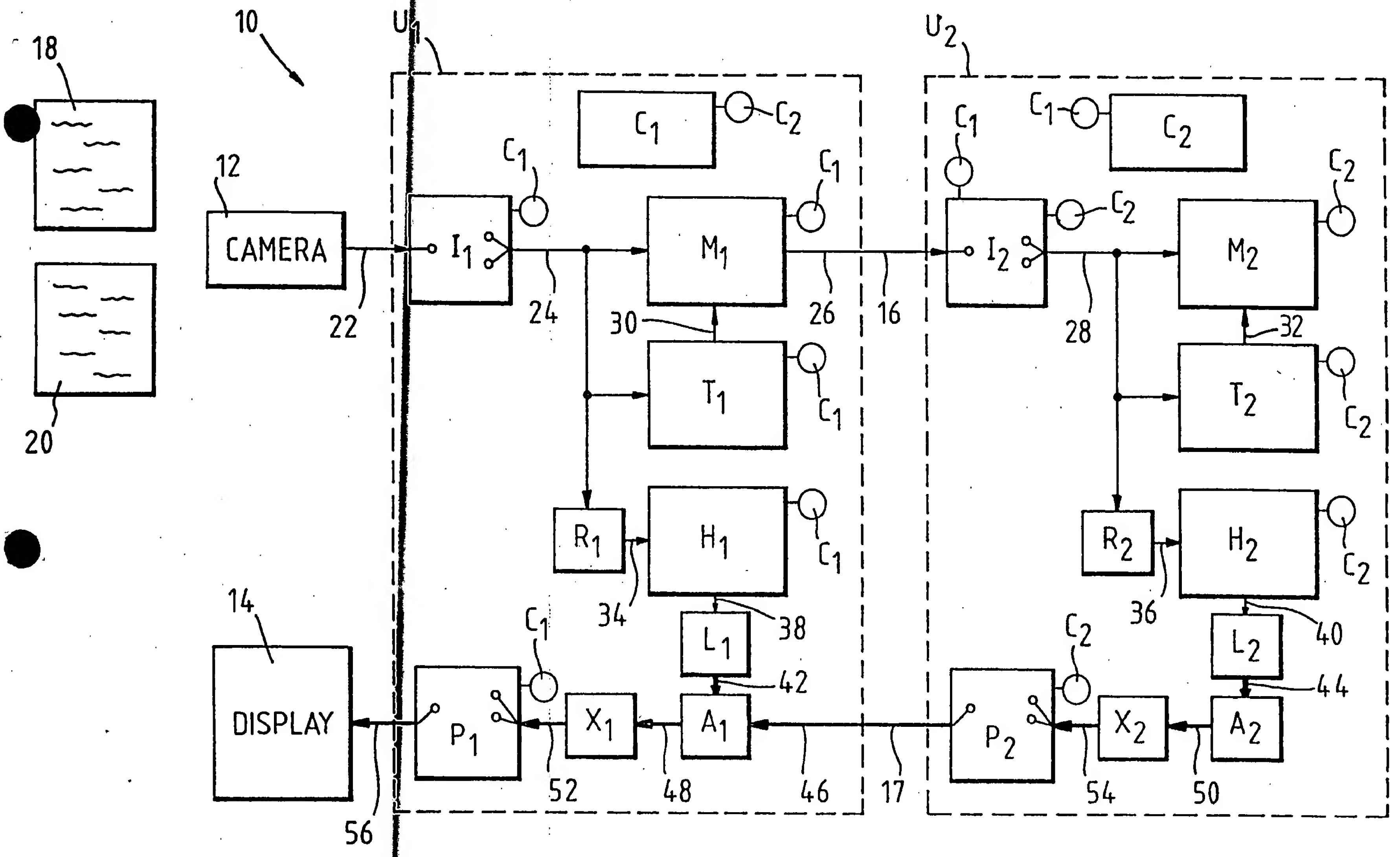
(57) An anomaly detector 10 comprises a camera 12, two modules U_1 and U_2 forming first and second stages of the detector respectively, and a display 14. The anomalies are highlighted through the use of a probability transformation, in which the value of an individual pixel is replaced by its occurence probability derived from other pixels associated during a training process. Prior to use the detector 10 is trained to detect anomalies in an image texture of interest. A training image 18 of the texture is viewed by camera 12 and stored. Pairs of pixels in the stored image are associated, after which their co-occurrence probabilities are determined and they undergo processing by training transputer T_1; the latter fills mapping look-up table M_1, which provides a data transformation for associated pixel signals prior to input to, and further association in the second module U_2. The training image 18 is replaced with another image 20 of the same texture, which is then processed by detector 10. This produces an anomaly image in which positions where anomalies occur in the image are indicated on the display 14.

Fig. 1.



At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

Fig.1.



10
18
20
12
CAMERA
22
14
DISPLAY
56
U1
U2
I_1
M_1
T_1
R_1
H_1
L_1
A_1
X_1
P_1
C_1
C_2
24
26
16
30
34
38
42
46
48
52
17
I_2
M_2
T_2
R_2
H_2
L_2
A_2
X_2
P_2
28
32
36
40
44
50
54

Fig. 2.

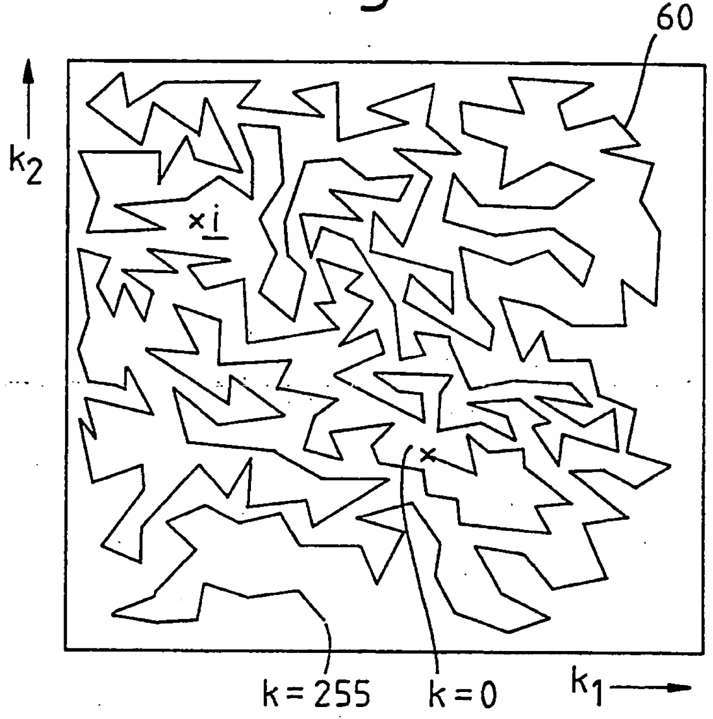


Fig. 3.

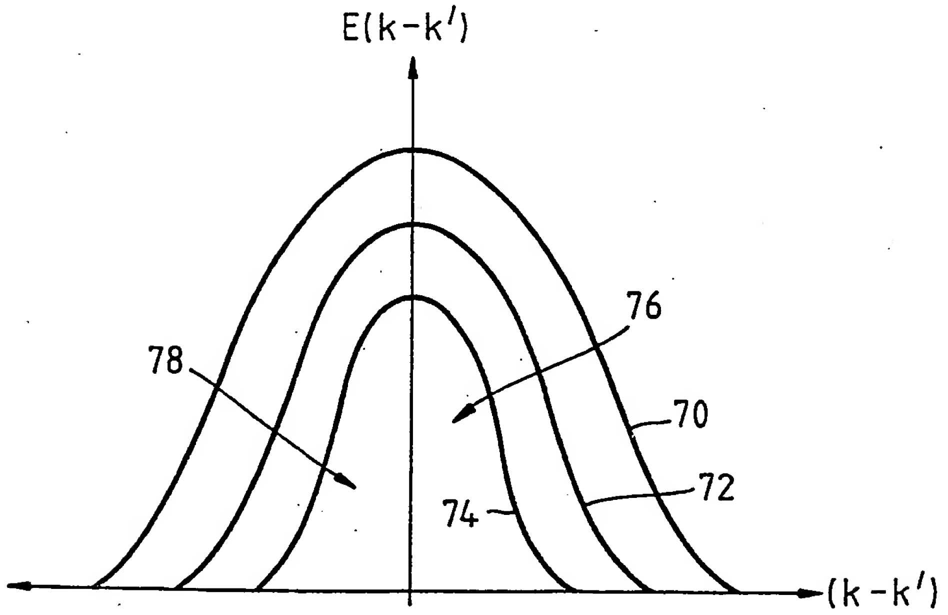


Fig. 4.

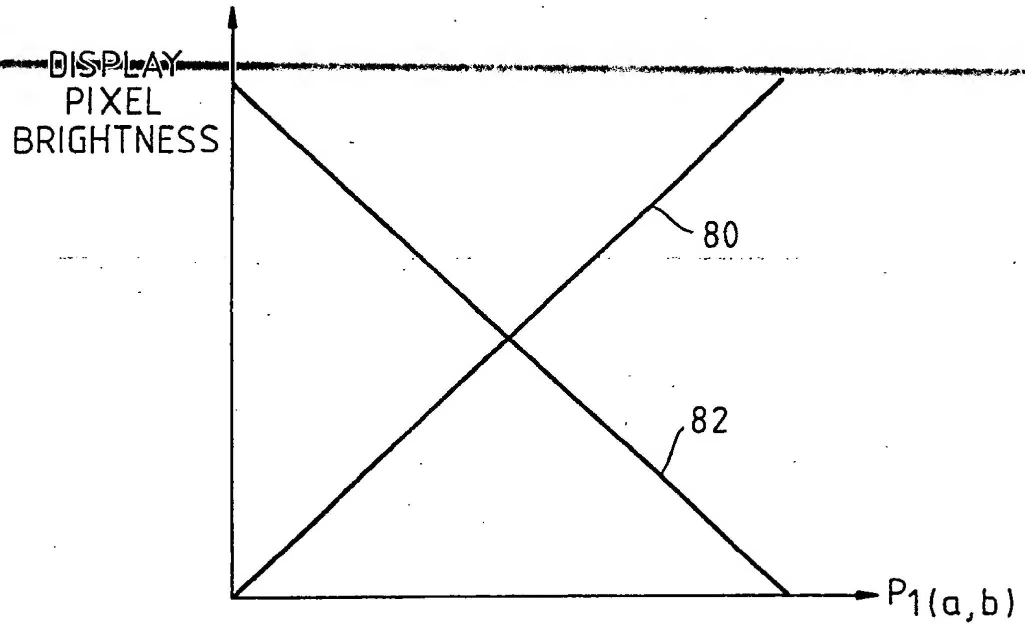
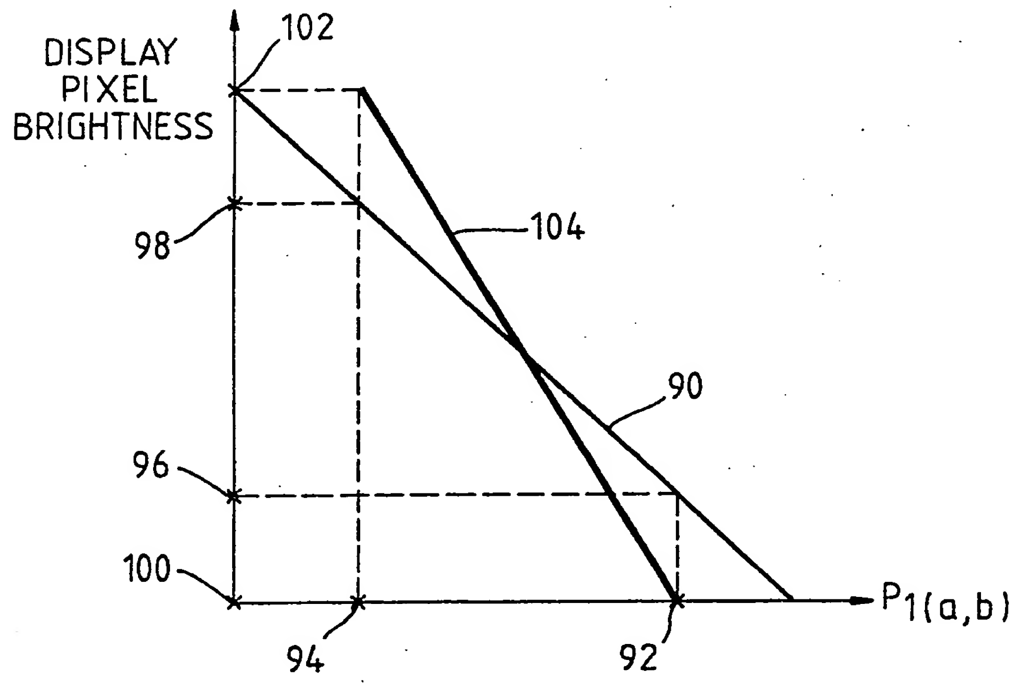
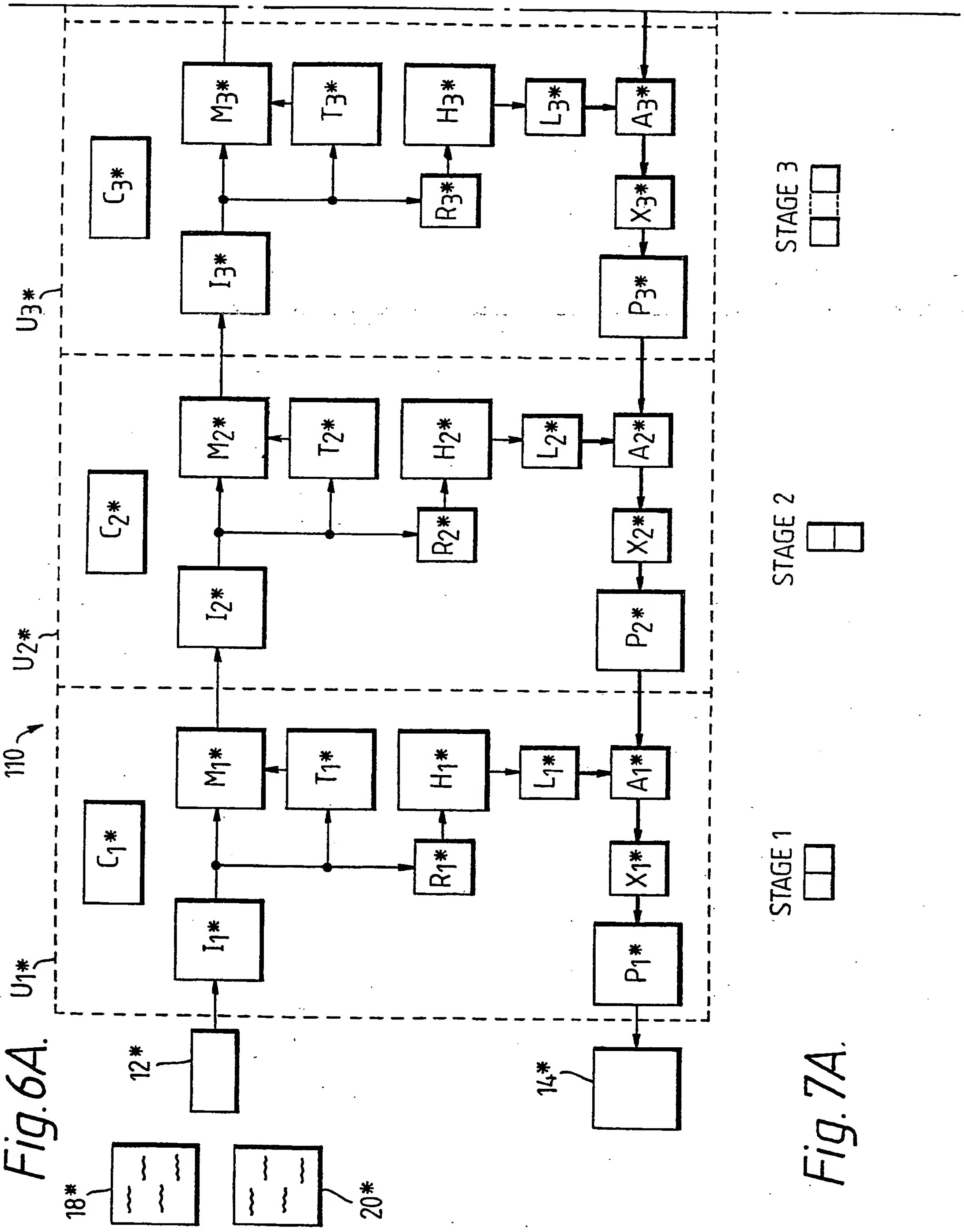

DISPLAY
PIXEL
BRIGHTNESS
80
82
P1(a,b)

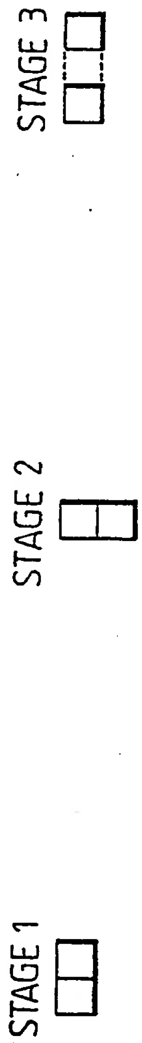
Fig. 5.


DISPLAY
PIXEL
BRIGHTNESS
102
104
98
90
96
100
94
92
P1(a,b)



U1*
U2*
U3*
110
C1*
C2*
C3*
I1*
I2*
I3*
M1*
M2*
M3*
T1*
T2*
T3*
R1*
R2*
R3*
H1*
H2*
H3*
L1*
L2*
L3*
A1*
A2*
A3*
X1*
X2*
X3*
P1*
P2*
P3*
12*
14*
18*
20*

Fig.6A.



STAGE 1
STAGE 2
STAGE 3

Fig.7A.

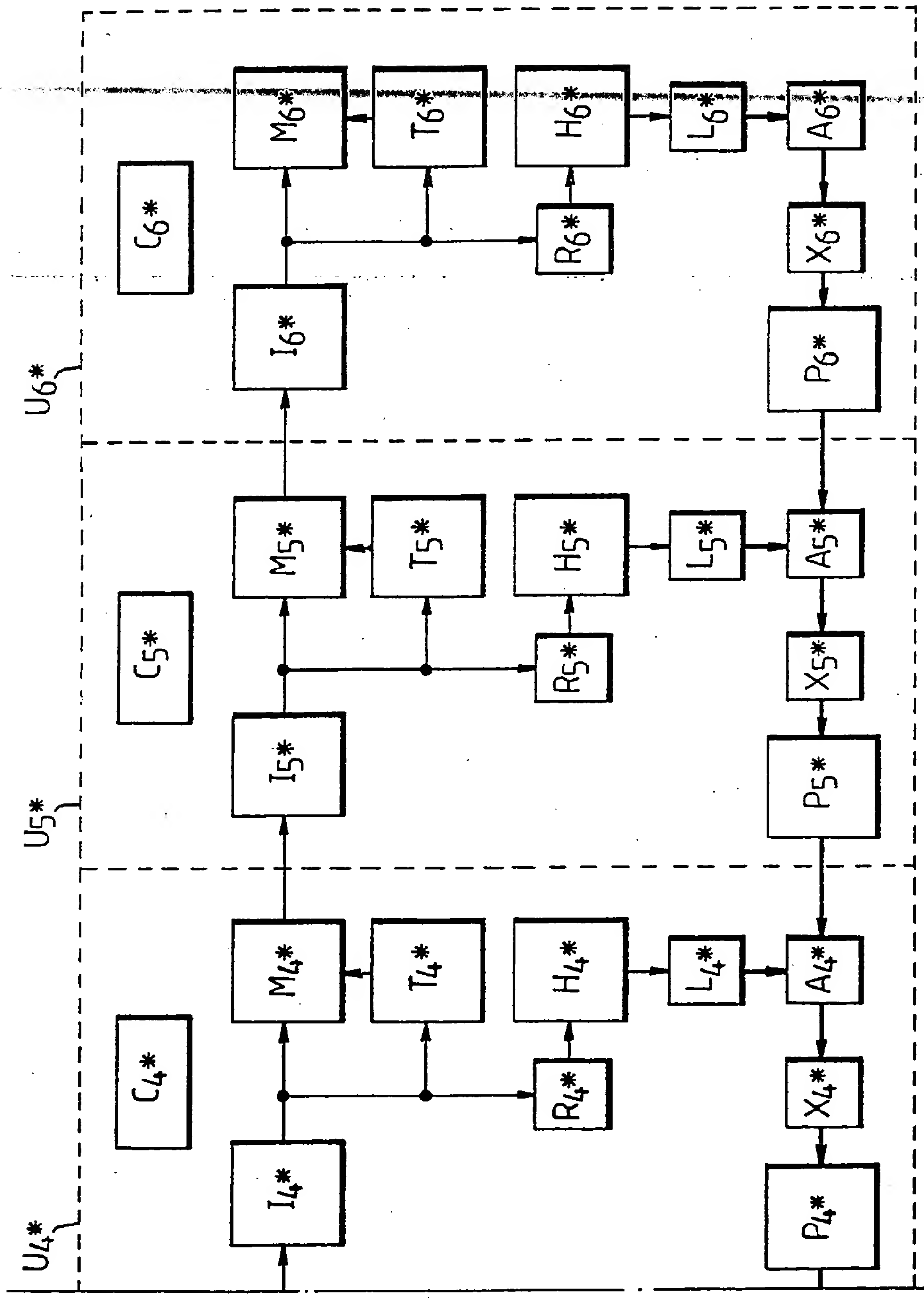

U4*
U5*
U6*
C4*
C5*
C6*
I4*
M4*
T4*
R4*
H4*
L4*
A4*
X4*
P4*
I5*
M5*
T5*
R5*
H5*
L5*
A5*
X5*
P5*
I6*
M6*
T6*
R6*
H6*
L6*
A6*
X6*
P6*

Fig.6B.

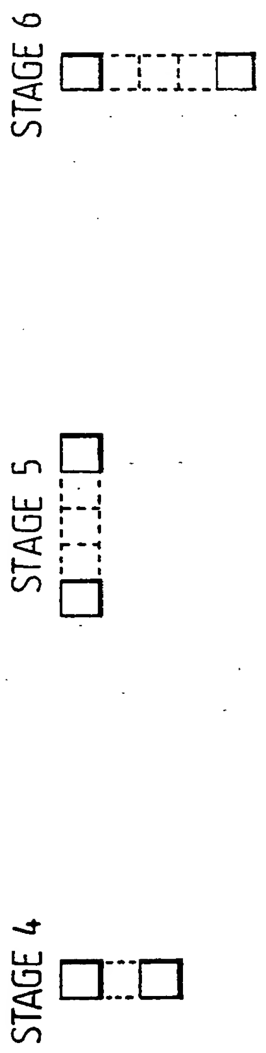

STAGE 4
STAGE 5
STAGE 6

Fig.7B.

Fig.8a)

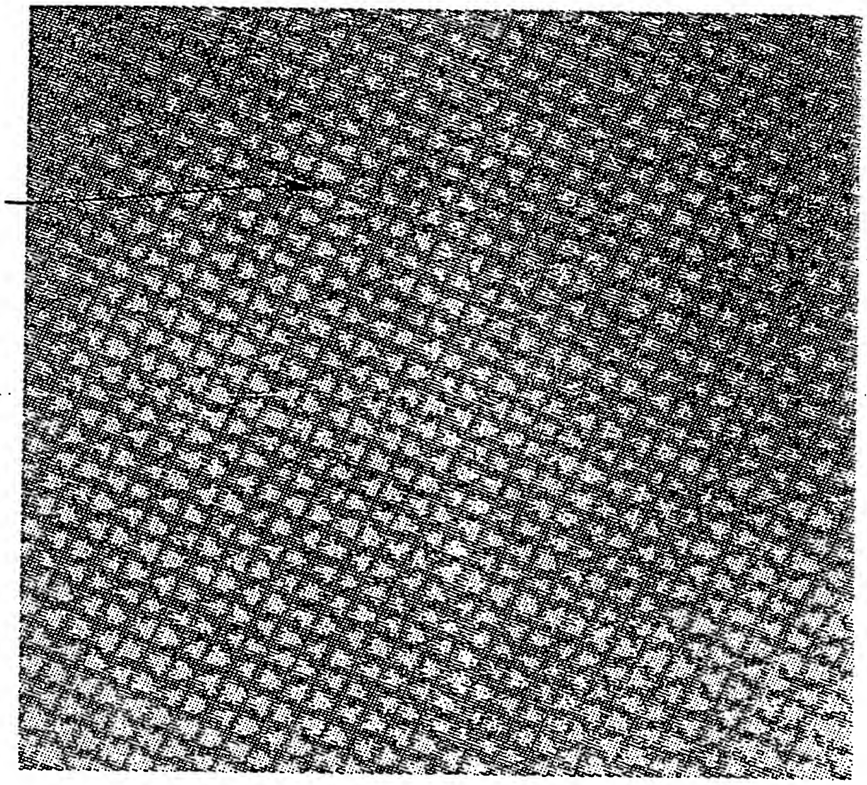


Fig.8b)

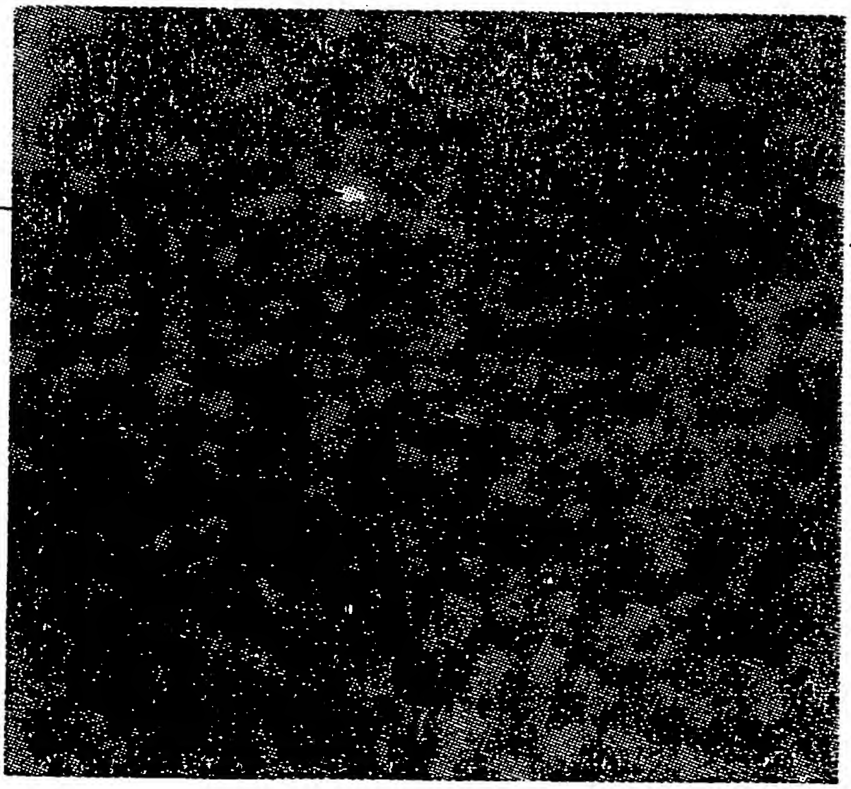


Fig. 9a)

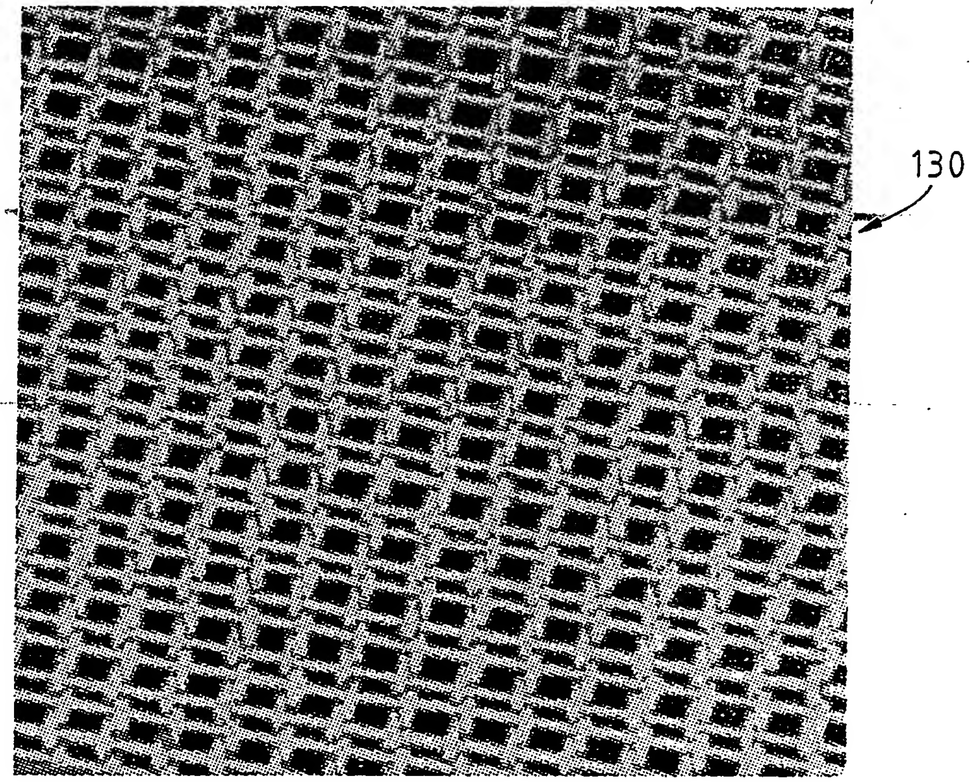


Fig. 9b)

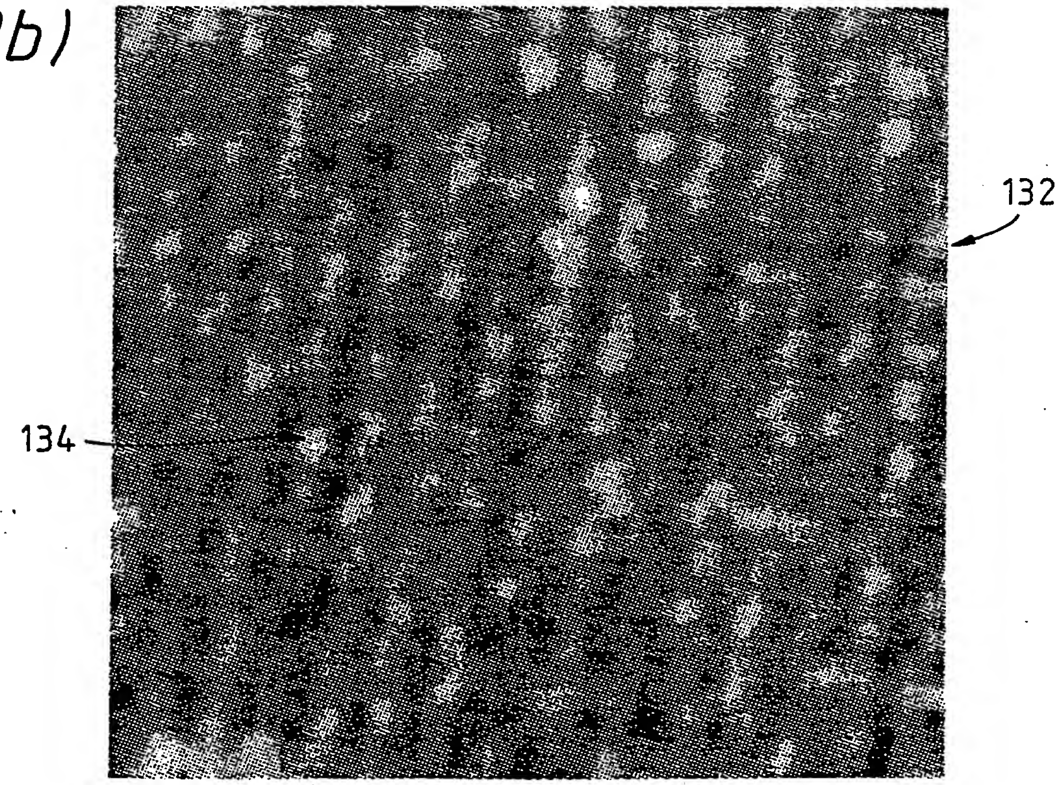


Fig. 10 a)

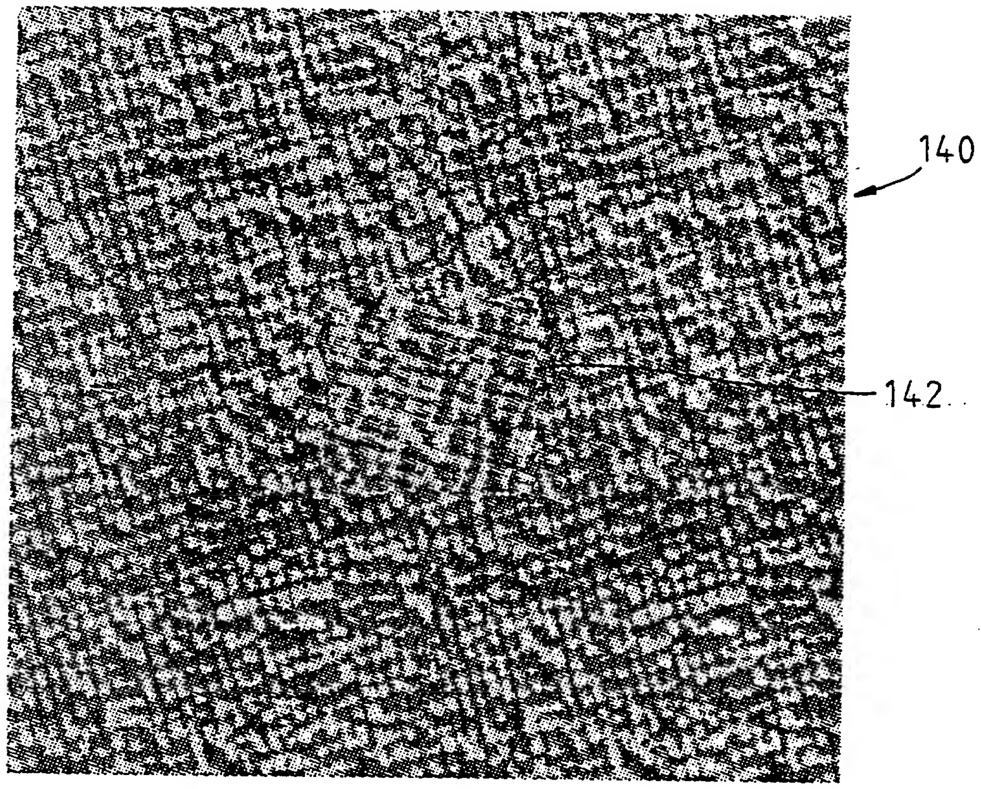
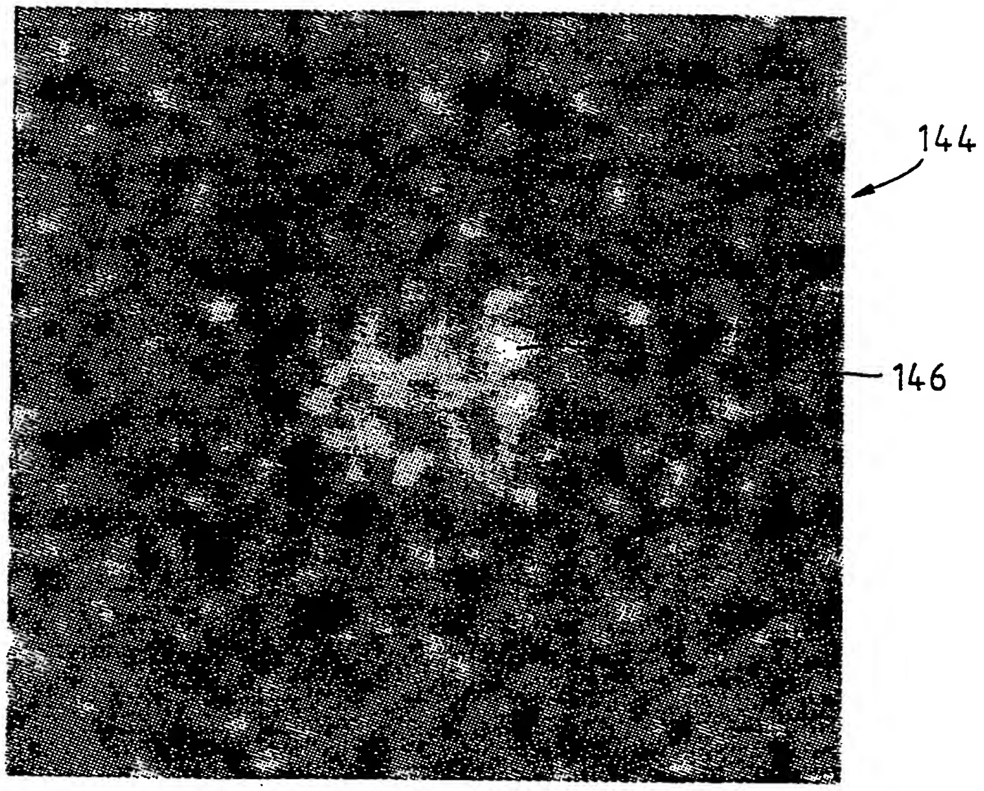


Fig. 10 b)



2253547

Anomaly Detector

This invention relates to an anomaly detector. More particularly, although not exclusively, it relates to such a detector for locating anomalies in an image of a texture. Relevant textures include woven fabrics with weave flaw anomalies and printed patterns with errors.

Flaws in the weave or knit of a fabric, and in the printing of patterns are at present difficult to detect during manufacture. The fabric or printed matter may need to be inspected by eye to ensure that substandard goods do not leave the factory. There is therefore a need for a device which may be arranged to inspect the goods during manufacture.

It is an object of the invention to provide an anomaly detector.

The present invention provides an anomaly detector including:

a) means for receiving input pixel signals,
b) means for associating pluralities of input pixel signals,
c) means for performing a probability transformation to replace associated input pixel signals by occurrence probability signals related thereto, and
d) means for accumulating a probability image comprising image pixel signals arising from functions of occurrence probability signals associated therewith.

The invention enables texture anomalies to be detected by providing data related to their occurrence probability. The invention provides, in the form of a probability transformation, statistical information on images of a texture. This information is related to the probability of pixel signals occurring with particular spatial relationships. When an image of the texture is tested for anomalies, pluralities of pixel signals occurring in a test image are associated and then transformed to occurrence probabilities using the probability transformation. A probability image is then accumulated from functions of occurrence probabilities, and it distinguishes between texture anomalies and more probable features.

The invention preferably includes:-

a) means for performing a data transformation on the associated input pixel signals to produce transformed pixel signals,
b) means for associating pluralities of transformed pixel signals to produce associated transformed pixel signals,
c) means for performing a probability transformation to replace associated transformed input pixel signals by occurrence probability signals related thereto, and
d) means for accumulating a probability image comprising image pixel signals arising from combining functions of occurrence probability signals related thereto and derived from input pixel signals with and without the said data transformation.

The data transformation may conveniently be a data compression transformation. The invention preferably includes trainable means for implementing the data transformation and the probability transformation.

The invention may include means for truncating associated pixel signals prior to probability transformation. The means for accumulating a probability image may include means for adding logarithms of probability transformed signals.

The means for performing a data transformation may be trainable to alter stored vectors in accordance with a function of pluralities of pixel signals and to implement a transformation by replacing pixel signals by respective codes indicating stored vectors associated therewith.

The anomaly detector of the invention may be arranged for investigation of progressively more complex texture spatial characteristics, and may include:-

a) transforming means for performing successive cumulative data transformations each involving transformation of respective associated pixel signals produced prior to that transformation and association of pluralities of transformed pixel signals produced by that transformation, the transforming means being arranged to produce successive sets of associated transformed pixel signals,

b) means for performing successive probability transformations to replace associated transformed pixel signals in each set by respective occurrence probability signals related thereto,

c) means for accumulating a probability image comprising image pixel signals arising from combining functions of occurrence probability transformed signals associated therewith and derived from a plurality of the sets of transformed pixel signals.

The means for performing successive cumulative data transformations may be trainable to alter sets of stored vectors in accordance with respective sets of associated pluralities of pixel signals and may implement data transformations in which successive sets of associated pixel signals are replaced by respective sets of codes indicating stored vectors associated therewith.

The means for accumulating a probability image may comprise successive accumulating means each arranged to compute a fraction (such as a quarter) of the sum of logarithms of functions of all occurrence probability signals arising subsequent to a respective data transformation added to the like of that occurrence probability signal arising immediately prior to that transformation, and to pass the result to a preceding accumulating means where available or to image storing means otherwise, each function being a respective occurrence probability signal which, at least in the case of signals arising subsequent to a first transformation, is divided by a product of summed row and summed column occurrence probabilities associated therewith.

The detector of the invention may be of modular construction, each module including:-

a) means for receiving input pixel signals,
b) means for associating pluralities of input pixel signals,
c) means for performing a probability transformation to replace associated input pixel signals by occurrence probability signals related thereto,
d) means for performing a data transformation on associated pixel signals to produce transformed pixel signals for output to a succeeding module of like kind to provide input pixel signals therefor,
e) means for accumulating probability image data comprising image pixel signal data arising from functions of occurrence probability signals associated therewith and derived from occurrence probability signals produced both in the respective module and a respective succeeding module where available, and
f) means for relaying probability image data to a preceding like module where available or to image retaining means otherwise.

In order that the invention might be more fully understood, embodiments thereof will now be described, by way of example only, with reference to the accompanying drawings, in which:-

Figure 1 schematically illustrates a two stage anomaly detector of the invention;

Figure 2 graphically illustrates part of a training procedure for the anomaly detector of Figure 1;

Figure 3 graphically illustrates a function E(k-k') used in the training procedure of Figure 2;

Figure 4	graphically illustrates two display schemes;
Figure 5	graphically illustrates the range of display pixel brightness values used;
Figure 6	schematically illustrates a six stage anomaly detector of the invention;
Figure 7	schematically illustrates pixels forming the 2-tuples for the six stages of the anomaly detector of Figure 5;
Figure 8	shows a) an optical image of a closed weave textile, and b) a corresponding anomaly image;
Figure 9	shows a) an optical image of an open weave textile, and b) a corresponding anomaly image; and
Figure 10	shows a) an optical image of a carpet, and b) a corresponding anomaly image.

Referring to Figure 1, a two stage anomaly detector of the invention is illustrated schematically. It is indicated generally by 10. The anomaly detector 10 includes a camera 12, first and second processing modules U, designated U_1 and U_2 respectively, and a display unit 14. The absence of a subscript index 1 or 2 indicates either or both modules U. The use of a subscript index n indicates a general nth module, in the case of the detector 10 n=1 or 2. Similar notation will be used for other units described later.

Each processing module U incorporates an image frame store I, a probability frame store P, a mapping look-up table M, a training transputer T and a truncation unit R. Each also incorporates a histogram microcomputer H, a log look-up table L, an addition unit A, a multiplication unit X and a control microprocessor C. The elements I, P, M, T, R, H, L, A, X and C have subscript indices 1 or 2 to indicate association with module U_1 or U_2 respectively. Each module U also includes interconnecting parallel data buses of eight, twelve, sixteen or thirty-two bits, as required in each position. In order to reduce illustrational complexity buses from the control microprocessors C_1 and C_2 are not illustrated. However, address and clock connections between the first control microprocessor C_1 and elements I_1, M_1, T_1, H_1, P_1 and C_2 are indicated by ringed indicia c_1. Similarly, address and clock connections between the second control microprocessor C_2 and elements I_2, M_2, T_2, H_2 and P_2 are indicated by ringed indicia c_2. Where illustrated eight, twelve and sixteen bit buses are indicated by narrow lines such as 16, and thirty-two bit buses are indicated by thick lines such as 17. To reduce illustration complexity the buses are not individually referenced. In the two stage anomaly detector 10 described below elements M_2, T_2, and A_2 in module U_2 are in fact redundant but are included to indicate the capability of extension to any number, N (subscript index n=1 to N), of modules U.

The anomaly detector 10 is intended inter alia for the detection of anomalies in image texture, for example faults in woven fabric. An anomaly is a feature in the image which has a low probability. The detector 10 first undergoes what is referred to as training; it processes a training image 18 of the texture of interest, the training image 18 having few, or no, anomalies. Subsequently it operates in a test mode to detect anomalies in images having the texture for which it has been trained.

Training may involve use of the training image 18 or a test image 20. Before describing the operation of the detector 10, in training and test modes, the various units which it incorporates, their interconnecting data buses and their individual operation in the detector 10 will be described.

The control microprocessors C are programmed to control the overall operation of the detector 10. Each includes a random number generator subroutine for pixel address generation during the training mode. Each supplies addresses and clock pulses, as appropriate, to the various elements I, T, M, H and P in the respective module U and to store I in the succeeding module U - if available - via respective connections (not shown but indicated in Figure 1 by c_1 and c_2 as appropriate). The operation of the control microprocessors C is therefore given in relation to each unit in the detector 10, in the following description, rather than as a whole.

In this specification a pair of pixels is referred to as a 2-tuple. It is also possible to employ higher order combinations of three or more pixels, which are referred to as 3-tuples, 4-tuples etc. The spatial relationship of a pair of pixels forming a 2-tuple is different in the different modules U of the detector 10; The overall set of 2-tuples used in a detector of the invention should be internally consistent, but sets other than that herein described may be used.

Each image frame store I stores images in the form of 256x256 pixel signals in a conceptual array having columns a=0 to 255 and rows b=0 to 255. Each pixel (a,b) stores an 8 bit (1 byte) signal. Image frame store I_1 has address and clock connections to the first control microprocessor C_1, together with an 8 bit input bus 22 from the camera 12, and a 16 bit output bus 24 to the mapping look-up table M_1, training transputer T_1 and truncation unit R_1. Image frame store I_2 has address and clock connections to the second control microprocessor C_2, together with an 8 bit input bus 26 from mapping look-up table M_1, and a 16 bit output bus 28 to the mapping look-up table M_2, training transputer T_2 and truncation unit R_2. In a generalised multi-stage equivalent of the detector 10, an nth-stage image frame store I_n would have an 8 bit input bus from mapping look-up table M_{n-1} and a 16 bit output bus to mapping look-up table M_n, training transputer T_n and truncation unit R_n. Each image frame store I is arranged such that in response to two pixel addresses, from the corresponding control microprocessor C, the respective two pixel signals, or 2-tuple contents, are output down the respective 16 bit output bus 24 or 28.

The training transputers T have 2 kbytes (1 kbyte = 1024 bytes) of memory, and are arranged to store 256 two-component vectors, each component being a 32 bit floating point number. The transputers are programmed with a training algorithm for the detector 10, to be described later. The training algorithm includes a random number generator subroutine. The first training transputer T_1 has address and clock connections to the first control microprocessor C_1, together with a 16 bit input bus 24 from image frame store I_1 and an 8 bit output bus 30 to the mapping look-up table M_1.

The second training transputer T_2 has address and clock connections to the second control microprocessor C_2, together with an 8 bit input bus 22 from the camera a 16 bit input bus 28 from image store I_2 and an 8 bit output bus 32 to the mapping look-up table M_2. In a generalised multi-stage device, transputer T_n has a 16 bit input bus from store I_n, and an 8 bit output bus 32 to mapping look-up table M_n. At the start of training each transputer T's internal memory is initialised with 256 random two-component vectors, each component being a 32 bit floating point number. Each vector is associated with a respective 8 bit index number k (0, 1, 2, ... or 255). During training each transputer T accepts 2-tuple contents from the corresponding image store I and processes them according to the training algorithm; the algorithm changes the values of the 256 vectors in the transputer's memory so that the vectors incorporate image 2-tuple information. Once a criterion set in the programme in each transputer T for the termination of training is met, the 256 vectors in its memory are frozen and are used to generate a transformation for the contents of the corresponding store I. At this stage the training mode is nearly complete and each transputer T then operates differently. All possible 2-tuple values (that is all possible 16 bit numbers) are then generated systematically and transformed according to the values in the transputer's memory established during training. The transformation consists of determining which of the vectors is closest to the input 2-tuple, and then outputting the index number k (0, 1, 2, ... or 255) of that vector; each index number has 8 bits, and is written to the location in the corresponding mapping look-up table M addressed by the respective 2-tuple contents. This is described in more detail later. The transputers T are not used in the test mode of the detector 10.

The mapping look-up tables M have 64 kbytes of memory arranged such that a 16 bit address addresses an 8 bit number. Mapping look-up table M_1 has address and clock connections to the first control microprocessor C_1, together with one 16 bit input data bus 24 from store I_1, one 8 bit input data bus 30 from transputer T_1 and one 8 bit output data bus 26 to store I_2, as previously mentioned in relation to those other units. Mapping look-up table M_2 has address and clock connections to the second control microprocessor C_2, together with one 16 bit input bus 28 from store I_2 and one 8 bit input bus 32 from transputer T_2. In a generalised N-stage device, the nth $(2<n<N)$ mapping look-up table M_n has one 16 bit input bus from store I_n, one 8 bit input bus from transputer T_n and one 8 bit output bus to store I_{n+1}. Initially mapping look-up tables M do not contain any data. During the training mode of the detector 10 mapping look-up table M_1 becomes filled with transformed values of all possible 2-tuple contents, ie a 2-tuple is replaced by the value of the index k of the respective associated vector stored in transputer T_1 after training, as described briefly above. In the test mode 2-tuple contents are accepted from store I_1 as addresses for locations in mapping look-up table M_1; the value of the index k stored in the location addressed are then read out and written to a pixel in store I_2. The address for the pixel in store I_2 is provided by control microprocessor C_1 and is related to the addresses of the 2-tuple from which the 2-tuple contents were read. Thus during the test mode mapping look-up table M_1 acts as a transformation, or mapping, look-up table providing a form of data compression transformation. The operation of mapping look-up table M_2 is very similar to that of mapping look-up table M_1, this will be discussed in more detail later.

Each truncation unit R is arranged to accept contents of a 2-tuple from the corresponding image frame store I as a 16 bit input. The 16 bit input is then split into the two 8 bit pixel signals of which it consists. The two 8 bit quantities are then truncated by removal of the two least significant bits of each. This forms two 6 bit quantities, which are combined to form a twelve bit output. Truncation unit R_1 has a 16 bit input bus 24 from image frame store I_1 and a 12 bit output bus 34 to histogram microcomputer H_1. Likewise truncation unit R_2 has a 16 bit input bus 28 from store I_2 and a 12 bit output bus 36 to microcomputer H_2. In general truncation unit R_n has a 16 bit input bus from store I_n and a 12 bit output bus to microcomputer H_n.

The histogram microcomputers H each have 8.25 (2{64x64+64+64}) kbytes of memory arranged to store 4096 histogram bins (in a conceptual array of 64 columns and 64 rows) and sums of column and row bin contents. Each bin contents is a 16 bit unsigned integer. The first histogram microcomputer H_1 has a 12 bit input data bus 34 from truncation unit R_1 and a 16 bit output data bus 38 to log look-up table L_1. Likewise the second histogram microcomputer H_2 has a 12 bit input bus 36 from truncation unit R_2 and an 8 bit output bus 40 to log look-up table L_2. In general histogram microcomputer H_n has a 12 bit input bus from truncation unit R_n and an 8 bit output bus to log look-up table L_n. The memories of the histogram microcomputers H are initialised with zeros, before receiving any input data. During the training mode each histogram microcomputer H operates such that a 12 bit input addresses a histogram bin, the first six bits addressing the correct row and the second six bits addressing the column, and one is added to the contents of that bin. Once the histogram is accumulated each bin is addressed, the contents read and its value checked. If a bin contents is zero then one is added to that bin. Next the sums of the bin contents of each row and column are computed and stored.

During the test mode each histogram microcomputer H operates such that a 12 bit input addresses a histogram bin, and the contents of that bin are read and are output to the corresponding log look-up table L, along with the appropriate row and column sums, down the respective 16 bit output bus. Thus the histogram microcomputers H perform a form of probability transformation.

Each log look-up table L incorporates a ROM plus a subtraction unit, shown in Figure 1 as one element L for illustrational simplicity. Each

ROM has 64k x 4 bytes of memory; a 16 bit address (64k) addresses a 32 bit (4 byte) number. Stored at each address is the natural log of that address, with the log of zero being set at zero. The logs of zero to $2^{16}-1$ are therefore stored as 32 bit floating point numbers. Log look-up table L_1 has a 16 bit input bus 38 from histogram microcomputer H_1 and a 32 bit output bus 42 to addition unit A_1. Likewise log look-up table L_2 has a 16 bit input bus 40 from histogram microcomputer H_2 and a 32 bit output bus 44 to addition unit A_2. In general log look-up table L_n has a 16 bit input bus from microcomputer H_n and a 32 bit output bus to addition unit A_n. Each log look-up table L is arranged such that in response to an address the log of that address is output. Thus when three values are output to a log look-up table L, from a histogram microcomputer H, three logs are read and passed within the log look-up table L to its subtraction unit. These logs are the log of the contents of a histogram bin, the log of the sum of the contents of the bins in the same row as that bin, and the log of the sum of the contents of the bins in the same column as that bin. In the subtraction unit of the second log look up table L_2, a calculation is carried out to subtract the second and third of these logs from the first. The result is passed down the 32 bit output bus to the respective addition unit A. Subtraction is not performed in the first module U_1 because of dimensional considerations governing the form of the required result.

The addition units A are arranged to accept two 32 bit floating point input numbers, sum them and output the 32 bit result. Addition unit A_1 has a first 32 bit input bus 42 from log look-up table L_1 and a second 32 bit input data bus 46 from probability image store P_2, and one 32 bit output bus 48 to multiplication unit X_1. Addition unit A_2 has a 32 bit input data bus 44 from log look-up table L_2 and a 32 bit output bus 50 to multiplication unit X_2. (In the detector 10 addition unit A_2 is in fact redundant; a 32 bit data bus could pass directly from log look-up table L_2 to multiplication unit X_2.) In a general N-stage device, an nth addition unit A_n $(2<n<N)$ would have a first 32 bit input bus from a log look-up table L_n, a second 32 bit input bus from a probability image store P_{n+1} in a succeeding module U_{n+1} and a 32 bit output bus to a multiplication unit X_n.

The multiplication units X are arranged to accept a 32 bit floating point input, multiply it by a factor F=0.25, and to output the 32 bit result. The first multiplication unit X_1 has a 32 bit input bus 48 from addition unit A_1 and a 32 bit output bus 52 to probability image store P_1. The second multiplication unit X_2 has a 32 bit input bus 50 from addition unit A_2 and a 32 bit output bus 54 to probability image store P_2. In an N-stage device an nth multiplication unit X_n ($2<n<N+1$) would have a 32 bit input bus from the corresponding addition unit A and a 32 bit output bus to the corresponding probability image store P.

The probability image stores P are initialised with zeros prior to any image data being written to them. Each probability image store P is used to store a probability image input to it from the respective multiplication unit X. The image is stored in the form of 256x256 pixels in a conceptual array having columns a=0 to 255 and rows b=0 to 255. Each stored pixel (a,b) is a 32 bit floating point quantity. Probability image store P_1 has a 32 bit input bus 52 from multiplication unit X_1 and a 32 bit output bus 56 to the display unit 14. Probability image store P_2 has a 32 bit input bus from multiplication unit X_2 and a 32 bit output bus to addition unit A_1. In general store P_n ($n>1$) has a 32 bit input bus from multiplication unit X_n and a 32 bit output bus to addition unit A_{n-1}. Each store P is arranged to accept a 32 bit quantity from the corresponding multiplication unit X, and to add it to the contents of two pixel addresses with which that quantity (related to a probability) is associated; the addresses are provided by the corresponding control microprocessor C. In the first module U_1, the two addresses are (a,b) and (a+1,b). A probability image is therefore accumulated in store P. In response to a pixel address from the corresponding control microprocessor C, a store P reads the contents of that pixel and outputs it down the corresponding 32 bit output bus. Store P_1 outputs the accumulated probability image to the display unit 14.

The procedure for training the anomaly detector 10 for a particular texture will now be described, with reference also to Figure 2 which illustrates part of the training procedure schematically.

The camera 12 views a training image 18 of the texture concerned, the image 18 being stored in the first frame store I_1. The first training transputer T_1 contains 256 two-dimensional vectors in its memory; the kth vector $\underline{v}_k$ (k=0,1,....255) has two 32 bit floating point scalar elements v_{k1} and v_{k2} which are inially set to random values. The vectors $\underline{v}_k$ are shown plotted in vector space in Figure 2. Each vector is a point and is connected to two adjacent vectors (where available) by respective line segments. Each vector therefore appears as a vertex between adjacent line segments, and the vectors and line segments collectively form a "string" 60 from k=0 to 255. The first control microprocessor C_1 is programmed to select a random 2-tuple of two adjacent pixels in the same row, that is a randomly chosen pair of adjacent pixels with addresses (a,b) and (a+1,b). The two pixels (a,b) and (a+1,b) are addressed in frame store I_1, and their intensities or pixel signals, $i_{1(a,b)}$ and $i_{1(a+1,b)}$ respectively, are read out to transputer T_1 as a vector $\underline{i}=(i_{1(a,b)},i_{1(a+1,b)})$. The vector $\underline{i}$ is also plotted in Figure 2 as "x". The Euclidean distance between each vector $\underline{v}_k$ and vector $\underline{i}$ is calculated by the first training transputer T_1. The vector $\underline{v}_k$ with the shortest Euclidean distance is designated $\underline{v}_{k'}$. This locates an origin k' about which a function E may be expanded. E is used to update each vector $\underline{v}_k$ stored in the transputer T_1 as described by the following expression:-

$$\underline{v}_k \Rightarrow \underline{v}_k + E(k-k').(\underline{i}-\underline{v}_k). \qquad (1)$$

The effect of updating according to this expression is to move each vector $\underline{v}_k$ towards the vector $\underline{i}$, in vector space, by a fraction of the distance between them represented by the vector $(\underline{i}-\underline{v}_k)$.

The fraction of $(\underline{i}-\underline{v}_k)$ added to the vector $\underline{v}_k$ is determined by the value of the function E at k. The value of the function E varies with the separation along the string 60 of a vector $\underline{v}_k$ from the vector $\underline{v}_{k'}$, that is with the value of (k-k'). It is at a maximum for k=k' and reduces as k moves away from k' along the string 60. The range of values that E can take is between zero and a prearranged maximum value designated g. Thus, when the vectors $\underline{v}_k$ are updated according to expression (1), vector $\underline{v}_{k'}$ has the greatest fraction of $(\underline{i}-\underline{v}_{k'})$ added and the further away a vector is along the string 60 the smaller the fraction of $(\underline{i}-\underline{v}_k)$ which is added.

The first control microprocessor C_1 then generates a further random address to read a second 2-tuple from the training image 18 stored in the first image store I_1, and the process of updating vectors $\underline{v}_k$ is repeated. Further 2-tuples are selected in the same way and vectors $\underline{v}_k$ updated until all 2-tuples have been used, and the training procedure is then terminated. There are a total of 256x(256-1) or 65280 2-tuples of the kind (a,b) and (a+1,b) in the first image store I_1. At the end of training, the vectors $\underline{v}_k$ are fixed at the values to which they have been transformed by repeated operation of Equation (1). They are subsequently used in the test mode, and remain fixed until the anomaly detector 10 is retrained.

Referring now also to Figure 3, function E(k-k') of expression 1 is illustrated graphically. During the training procedure the function E(k-k') is adjusted. That is, in this embodiment, the function E is of the form $g.\exp[-f(k-k')^2]$ and the values of the parameters f and g are adjusted during training so that later vectors have progresssively less effect. This is shown in Figure 3 by curves 70, 72 and 74 and emphasised by the arrows 76 and 78. The effect of the increase in the value of parameter f is to narrow the function E, and thus to reduce the number of vectors $\underline{v}_k$ on either side of $\underline{v}_{k'}$ which are updated to a significant extent. The effect of the reduction in the value of parameter g is to reduce the height of the function E, and thus to reduce the fraction of $(\underline{i}-\underline{v}_k)$ added to $\underline{v}_k$. Each adjustment of parameters f and g occurs after a preset number of vector updates has been performed. The number of times the parameters f and g are adjusted is also preset. The values required to control the training procedure in this way are stored in transputer T_1. The training procedure is terminated after the preset number of vector updates has been performed following the final adjustment of parameters f and g. The training procedure is described in more detail by T Kohonen in "Self-Organization and Associative Memory", 3rd edition, Springer Verlag (1989).

Next the first mapping look-up table M_1 is filled as follows. Control microprocessor C_1 sequentially generates all possible 16 bit numbers. These are used as all possible vectors $\underline{i}$ in transputer T_1 and as all possible 16 bit addresses in mapping look-up table M_1. Each vector $\underline{i}$ so generated is presented to the training transputer T_1, where the closest of vectors $\underline{v}_k$ to it is calculated, this vector being designated $\underline{v}_{k'}$. The respective 8 bit value of the index number k' of that vector is then passed to mapping look-up table M_1, where it is stored in the location addressed by the respective 16 bit address. When filled, the table M_1 contains 8 bit index numbers k associated with 2x32 bit pixel signals and providing a data compression transformation.

The next stage in the training mode is to map the contents of the first frame store I_1 into the second frame store I_2, using the mapping look-up table M_1, and to fill the memory of the histogram microcomputer H_1. First the memory of microcomputer H_1 is initialised with zeros. Next control microprocessor C_1 sequentially addresses 2-tuples in frame store I_1 until all 2-tuples of the form (a,b), (a+1,b) have been addressed. Thus the index a is incremented from 0 to 254 with b=0, then with b=1, and so on until b=255 has been completed. The index a cannot be incremented to 255 because the value of a+1 must be within the limits (0 to 255) of the 256x256 frame store I_1. When a 2-tuple (a,b), (a+1,b) is addressed the signals $i_{1(a,b)}$ and $i_{1(a+1,b)}$ are read and concatenated to form a 16 bit address. The 16 bit address is applied to the mapping look-up table M_1; the contents of the location addressed are then read and written to pixel (a,b) in frame store I_2. The pixel signals $i_{1(a,b)}$ and $i_{1(a+1,b)}$ also pass to truncation unit R_1 where the two least significant bits of each of $i_{1(a,b)}$ and $i_{1(a+1,b)}$ are removed. The reason for the truncation of the pixel signals will be explained later. The two 6 bit quantities are then concatenated to form a 12 bit address, with truncated $i_{1(a,b)}$ forming the 6 most significant bits. The 12 bit address is applied to microcomputer H_1, and it addresses a memory location within the microcomputer H_1 referred to as a "bin"; one is then added to the contents of the bin addressed. This procedure is followed for all 2-tuples in store I_1. The bins collectively form a histogram accumulated in the memory of the microcomputer H_1.

Each bin in the histogram in the first microcomputer H_1 contains a respective number indicating how many 2-tuples in store I_1 have intensities given by the relevant address. In addition each pixel in store I_2, except those in column a=255, contains the result of transforming a 2-tuple from store I_1 into an index number k associated with a vector $\underline{v}_k$ closest to the relevant 2-tuple pixel intensities. The complete histogram has thus been accumulated.

The histogram accumulated in microcomputer H_1 consists of 64x64 bins, in conceptual columns with indices c = 0 to 63 and rows with indices d = 0 to 63; the general bin at address (c,d) has bin contents designated $h_{1(c,d)}$. As has been said, addresses are formed by truncation of pixel contents in the truncation unit R_1. Truncation has a beneficial effect on errors in bin contents and on uncertainty in the probability transformation stored in the histogram. The error in a bin contents is the square root of the contents. Thus with the reduced number of histogram bins containing higher counts the relative errors in the contents are smaller.

The next step is for the sum S_{1c} of bin contents $h_{1(c,d)}$ in each column c and the sum S_{1d} in each row d (c, d = 0 to 63) are calculated and stored in the memory of microcomputer H_1, as follows:-

$$S_{1c} = {}_{d}\, h_{1(c,d)} \qquad (2)$$

$$S_{1d} = {}_{c}\, h_{1(c,d)} \qquad (3)$$

The next stage is for the information stored in the second image store I_2 to be processed in the second module U_2. It contains vector references k each associated with two respective row neighbour pixels and each stored in the second image store I_2 as though it were the contents of a pixel in the first image store I_1. Pairs of column neighbour references are addressed in the second image store I_2 to form 2-tuples each associated with four pixels in the first image store I_1. This is carried out as follows. The memory of the second histogram microcomputer H_2 is initialised with zeros, and a histogram is accumulated within it. The second control microprocessor C_2 systematically selects 2-tuples from the second image store I_2 until all possible column neoghbour 2-tuples of the form (a,b), (a,b+1) have been addressed. To do this, index a is incremented from a=0 to 254 with b=0, then with b=1 and so on until b=254 has been completed. The index a is not incremented to 255 because the a=255 column of the second store I_2 was not addressed when the image in the first store I_1 was transformed and mapped to store I_2, and therefore there are no values stored in that column. The index b cannot be incremented to 255 because the value of b+1 must be within the limits of the 256x256 frame store I_2. When a 2-tuple (a,b), (a,b+1) is addressed the signals $i_{2(a,b)}$ and $i_{2(a,b+1)}$ are read, pass to truncation unit R_2 where they are truncated, and concatenated to form a 12 bit address, as described above. The 12 bit address is applied to the memory of microcomputer H_2 and one is added to the contents of the bin addressed. This procedure is followed until all 2-tuples in store I_2 have been processed, and thus the histogram accumulation has been completed.

The histogram accumulated in microcomputer H_2 consists of 64x64 bins, in columns c c= 0 to 63 and rows d = 0 to 63, with bin contents $h_{2(c,d)}$. The sums S_{2c} and S_{2d} of bin contents $h_{2(c,d)}$ in each column c and row d respectively are calculated and stored in the memory of microcomputer H_2 as follows:-

$$S_{2c} = {}_{d}\, h_{2(c,d)} \qquad (4)$$

$$S_{2d} = {}_{c}\, h_{2(c,d)} \qquad (5)$$

The training of the anomaly detector 10 for the texture of image 18 is now complete and the training transputer T_1 and the mapping look-up table M_1 are not used again until the anomaly detector 10 is retrained. The anomaly detector 10 is now ready to process images of the texture for which it has been trained and in which anomalies are to be detected. The training image 18 may be replaced with an image 20 of the same texture as that of the training image 18, or alternatively the training image 18 may now itself be processed. The description that follows assumes that the training image has been replaced by image 20.

In the embodiment of Figure 1 employing only two modules U_1 and U_2, it is not necessary to train the second training transputer T_2 and mapping look-up table M_2; such training would only be needed if there were a third module.

The image 20 to be processed is viewed by the camera 12 and is stored in frame store I_1. The image stored in store I_1 is then transformed and mapped to store I_2 as follows. Control microprocessor C_1 sequentially addresses all 2-tuples of the form (a,b), (a+1,b) in frame store I_1, reads the signals $i_{1(a,b)}$ and $i_{1(a+1,b)}$ and concatenates them to form a 16 bit address. The address is applied to the mapping look-up table M_1, the contents of the location addressed are read and written to pixel (a,b) of frame store I_2.

Next "probability images" are accumulated in the first and second stores P_2 and P_1 using the image information stored in stores I_2 and I_1 and the histograms stored in microcomputers H_2 and H_1. The first step is for the second control microprocessor C_2 to initialise all store P_2 with zeros; subsequently, the second control microprocessor C_2 systematically addresses all 2-tuples of the form (a,b), (a,b+1) in the second store I_2 as described above; the pixel signals are read from the store I_2, and are truncated and concatenated before being passed to the second microcomputer H_2 as a 12 bit address. The contents $h_{2(c,d)}$ of the bin (c,d) addressed in microcomputer H_2 are read and pass to the second log look-up table L_2 as an address.

The sums S_{2c} and S_{2d} of the corresponding column c and row d stored in the second microcomputer H_2 are also read and used to address the second log look-up table L_2; This generates the logs of the addresses, which pass to the subtraction unit within the log look-up table L_2. Here a quantity defined as Q_2 is calculated as follows:-

$$Q_2 = \log(h_{2(c,d)}) - \log(S_{2c}) - \log(S_{2d}) \qquad (6)$$

Q_2 is passed to the second multiplication unit X_2 where it is multiplied by a factor $F=1/2^2=0.25$. The resulting Q_2/F is then added to the contents of both pixel (a,b) and pixel (a,b+1) in store P_2.

A similar process is now carried out in the first module U_1 of the anomaly detector 10. The first step is for the first store P_1 to be initialised with zeros. Then the first control microprocessor C_1 sequentially addresses all pixels of the form (a,b), (a+1,b) in the first store I_1, as previously described. For each 2-tuple the pixel signals $i_{1(a,b)}$ and $i_{1(a+1,b)}$ are read, truncated, concatenated and pass to the first microcomputer H_1 as a 12 bit address. The contents $h_{1(c,d)}$ of the bin (c,d) addressed in microcomputer H_1 are read and pass to the first log look-up table L_1 as an address. The sums S_{1c} and S_{1d} of column c and row d are also read and passed to log look-up table L_1 as addresses. The contents of the locations addressed in log look-up table L_1 are read and pass to the subtraction unit within it, where a calculation of Q_2 equivalent to that of Equation (6) is performed. The result of the calculation passes to the first addition unit A_1. The first control microprocessor C_1 supplies to the second such microprocessor C_2 pixel address (a,b) in frame store P_2. The second control microprocessor C_2 consequently addresses that pixel, and its contents $p_{2(a,b)}$ are read and pass to addition unit A_1, which adds them to the quantity passing from log look-up table L_1. This sum passes to the first multiplication unit X_1 where it is multiplied by a factor

F=0.25 before being added to the contents of both pixel (a,b) and pixel (a+1,b) in the first store P_1. This procedure is carried out for all pixels. At this point, the first store P_1 contains probability information on 2-tuple occurrence (from U_1) combined with the like on 2-tuple of 2-tuple occurrence (from U_2).

Strictly speaking, a calculation of Q_1 equivalent to that of Equation (6) (but involving sums S_{1c} and S_{1d}) is not in fact required in the first module U_1 because of probability dimensionality considerations. Q_1 should theoretically be set equal to $\log(h_{1(c,d)})$. However, use of an equivalent of Equation (6) in the first module U_1 does not affect the applicability of the result. Such calculations are required the second module U_2, and it is convenient for the corresponding histogram microcomputers H_1 and H_2 and log look-up tables L_1 and L_2 to have like programming. Moreover, in an N stage device of the invention, such calculations would be required in the second to Nth modules.

Referring now to Figure 4, two display schemes are illustrated graphically. These schemes are for use in displaying results obtained by the detector 10. A graph 80 illustrates a scheme in which pixel brightness increases linearly with increasing pixel contents value. This is referred to as a "probability image" because the more likely the contents values of 2-tuples being processed the brighter the respective pixel in the displayed "probability image". A graph 82 illustrates a scheme in which pixel brightness has an inverse linear relationship to pixel contents value. This is referred to as an "inverse probability image" or an "anomaly image", because in this image brighter pixels correspond to less probable contents values of 2-tuples being processed. This latter scheme makes it easier to identify anomalies in a displayed image, as will be described later. It should be noted that non-linear display schemes may also be employed.

Referring now also to Figure 5, the range of display pixel brightness values obtained using the scheme of graph 82 is illustrated graphically. Graph 90 is the equivalent of graph 82. Points 92 and 94 indicate the upper and lower limits respectively of values of $p_{1(a,b)}$, in store P_1. Points 96 and 98 indicate the lower and upper limits respectively of display pixel brightness, corresponding to the limits of $p_{1(a,b)}$ indicated by points 92 and 94. Points 100 and 102 indicate the lower and upper limits of display pixel brightness possible on the display 14. Thus, if the image stored in store P_1 is displayed as illustrated in Figure 5, a significant portion of the range of display pixel brightness is not being used. Therefore, in order to improve the visibility of anomalies in the displayed image, the display scheme is modified, as illustrated by graph 104, such that the complete range of display pixel brightness is used. Thus the lower limit of $p_{1(a,b)}$, point 94, is mapped onto the upper limit of display pixel brightness, point 102. Similarly the upper limit of $p_{1(a,b)}$, point 92, is mapped onto the lower limit of display pixel brightness, point 100.

The "probability image" stored in store P_1 is then displayed on the display 14, using the display scheme of graph 104, thus providing an "anomaly image". Anomalies in the image 20 being processed are necessarily of low probability; they therefore show up on the display 14 as bright areas. Thus faults in knitted or woven fabric, paper or any other such texture may be detected. Also errors in printing of, for example, fabrics, wallpaper, tiles and artificial work surface materials may be detected.

The anomaly detector 10 described thus far is a two stage embodiment. In general, anomaly detectors may be constructed to have N stages. An N stage detector has N modules such as U in Figure 1. A six stage (N=6) anomaly detector will now be described.

Referring now to Figure 6, a six stage anomaly detector 110 of the invention is illustrated schematically. Parts equivalent to those described with reference to Figure 1 are like referenced with the addition of an asterix (*) index, the * being on the subscript if present. Generally the notation is as used in the description of detector 10 of Figure 1. When referring generally to any part which occurs in each of the N stages, and not to a specific one of these parts, it is denoted with a subscript n*. The six stage anomaly detector 110 includes six modules U*, designated U_{1*}, U_{2*}, U_{3*}, U_{4*}, U_{5*}, and U_{6*}. Modules U_{1*} to U_{6*} are the first to sixth stages of the detector 110 respectively. For simplicity of illustration all connections between control microprocessors C_{1*} to C_{6*} and other parts of the six stage detector 110 have been omitted from Figure 6. The detector 110 also includes a camera 12*, and a display unit 14* similar to the respective elements described for the detector 10. In the six stage detector 110 elements M_{6*}, T_{6*} and A_{6*} are redundant. As before they are included to indicate the capability of extension to greater than N=6 stages.

Referring now also to Figure 7, 2-tuples used in each stage of the six stage anomaly detector 110 are illustrated below the respective stage in Figure 5. Training and operation of the detector 110 are very similar to those of the detector 10. Prior to training, the memories of the histogram microcomputers H are initialised with zeros. The training of the first stage U_{1*} is exactly as described for the two stage detector 10, using row nearest neighbour 2-tuples (a,b), (a+1,b). It is completed when the mapping look-up table M_{1*} has been filled, the contents of store I_{1*} have been transformed and mapped to store I_{2*}, and a histogram has been accumulated in the memory of microcomputer H_{1*} along with the sums of bin contents for each column and row of the histogram.

In the six stage detector 110, once the second image store I_{2*} is filled, its contents are processed in the second stage U_{2*} which is trained using column nearest neighbour 2-tuples (a,b), (a,b+1). The training of the second stage U_{2*} is complete when the mapping look-up table M_{2*} has been filled, the contents of store I_{2*} transformed and mapped to store I_{3*} and a histogram accumulated in the memory of microcomputer H_{2*} along with the sums of bin contents of each column and row of the histogram.

The contents of the third image store I_{3*} are processed in the third stage U_{3*}, which is trained using row second nearest neighbour 2-tuples (a,b), (a+2,b). As before this is complete when mapping look-up table M_{3*} has been filled, the contents of store I_{3*} transformed and mapped to store store I_{4*} and a histogram accumulated in the memory of microcomputer H_{3*} along with the sums of bin contents of each column and row of the histogram. When reading the contents of store I_{3*} in 2-tuples (a,b), (a+2,b) the index a is incremented from 0 to 252 and b from 0 to 254.

The contents of store I_{4*} are then processed in the fourth stage U_{4*}, which is trained using column second nearest neighbour 2-tuples (a,b), (a,b+2). The training of this stage is complete when mapping look-up table M_{4*} has been filled, the contents of store I_{4*} transformed and mapped to store I_{5*} and a histogram accumulated in the memory of microcomputer H_{4*} along with the sums of bin contents in each column and row of the histogram. When reading the contents of store I_{4*} in 2-tuples (a,b), (a,b+2) index a is incremented from 0 to 252 and b from 0 to 252.

The contents of store I_{5*} are processed in the fifth stage U_{5*}, which is trained using row fourth nearest neighbour 2-tuples (a,b), (a+4,b). As in preceding stages, the training is complete when the mapping look-up table M_{5*} has been filled, the contents of store I_{5*} transformed and mapped to store I_{6*} and a histogram accumulated in the memory of microcomputer H_{5*} along with the sums of bin contents in each column and row of the histogram. When reading the contents of store I_{5*} in 2-tuples (a,b), (a+4,b), index a is incremented from 0 to 248 and b from 0 to 252.

The contents of store I_{6*} are processed in the sixth stage U_{6*}, which is trained using column fourth nearest neighbour 2-tuples (a,b), (a,b+4). Since the sixth stage U_{6*} is the final stage of the detector 110, as in module U_2 of the detector 10, the transputer T_{6*} and mapping look-up table M_{6*} are redundant; they are included to demonstrate use of like modular stages.

The sixth stage U_{6*} requires training only as regards accumulation of a histogram in the memory of microcomputer H_{6*} along with the sums of bin contents of each column and row of the histogram. When reading the contents of store I_{6*} in 2-tuples (a,b), (a,b+4), index a is incremented from 0 to 248 and the index b from 0 to 248. The training transputers T_{1*} to T_{5*}, in modules U_{1*} to U_{5*} respectively, are not used again until the detector 110 is retrained.

The training of the six stage detector 110 is carried out using a training image 18 (as described with reference to Figure 1). For use of the detector 110 in test mode, the training image is replaced with a test image 20. The test image 20 is viewed by a camera (not shown) and stored in the first frame store I_{1*}. The contents of store I_{1*} are then transformed, using the transformation stored in mapping look-up table M_{1*}, and mapped to store I_{2*}. Like transformations and mappings occur between each successive pair of image stores I_{2*}/I_{3*}, I_{3*}/I_{4*}, I_{4*}/I_{5*}, and I_{5*}/I_{6*}. The transformation between the nth pair of image stores I_{n*}/I_{n+1*} (n = 1 to 5) employs the transformation stored in the nth mapping look-up table M_{n*}.

As in the two stage detector 10 "probability images" are now accumulated in reverse order using the images stored in stores I_{6*} to I_{1*} and histograms stored in microcomputers H_{6*} to H_{1*}. The process of constructing the probability images starts in the sixth stage U_{6*} of the detector 110 and works back to the first stage U_{1*}. It should be noted that each probability image store P* is initialised with zeros by the respective control microprocessor C* prior to any image data being written to it.

The sixth control microprocessor C_{6*} selects pixels of the form (a,b), (a,b+4) in store I_{6*}. For each 2-tuple the pixel contents $i_{6(a,b)}$ and

$i_{6(a,b+4)}$ are read, truncated, concatenated and pass to microcomputer H_6. as a 12 bit address. The contents of the bin addressed in microcomputer H_6. are read and pass to log look-up table L_6. as an address. The sums of the corresponding column and row are also read and pass to log look-up table L_6. as addresses. The contents of the locations addressed in log look-up table L_6., the logs of the addresses, are read and pass to the subtraction unit within log look-up table L_6.. A calculation equivalent to that of Equation (6) is carried out and the result passes to multiplication unit X_6. where it is multiplied by a factor F=0.25. The quantity thus calculated is then added to both pixel (a,b) and pixel (a,b+4) in store P_6.. The process is repeated until all 2-tuples of the form (a,b), (a,b+4) in store I_6. have been addressed.

The equivalent process is now carried out in the fifth stage U_5.. Control microprocessor C_5. selects 2-tuples of the form (a,b), (a+4,b) in store I_5.. For each 2-tuple the pixel contents $i_{5(a,b)}$ and $i_{5(a+4,b)}$ are read, truncated, concatenated and pass to microcomputer H_5. as a 12 bit address. The contents of the bin addressed in microcomputer H_5. are read and pass to log look-up table L_5. as an address. The sums of the corresponding column and row are also read and pass to log look-up table L_5. as addresses. The contents of the locations addressed in log look-up table L_5. are read and a calculation equivalent to that of Equation (6) is carried out. The result of the calculation passes to addition unit A_5.. The contents of pixel (a,b) in store P_6. are read and also pass to addition unit A_5. where they are added to the quantity passing from log look-up table L_5.. Their sum is multiplied by a factor F=0.25 in multiplication unit X_5. and is then added to pixel (a,b) and pixel (a+4,b) in P_5.. The process is repeated until all 2-tuples of the form (a,b), (a+4,b) in store I_5. have been addressed.

The process described above for the fifth stage U_5. is repeated back through each of the fourth, third, second and first stages, U_4. to U_1., of the detector 110. In each stage the 2-tuple form employed is that used when training that stage. Consequently, the first stage store P_1 receives probability information on 2-tuple occurrence from U_1. combined with the like on 2-tuple of 2-tuple occurrence from U_2., 2-tuple of

2-tuple of 2-tuple occurrence from U_{3*} and so on up to U_{6*}. The first store P_1 therefore contains information on the probability of occurrence of a variety of pixel geometries. The contents of store P_{1*} are then displayed on an equivalent (not shown) of display 14 using the display scheme of graph 104, thus providing an "anomaly image".

The two stage anomaly detector 10 and the six stage anomaly detector 110 operate by measuring the statistical properties, specifically co-occurrence matrices, of coded versions of an input test image. That is, in the anomaly detector 10 the image stored in the second image store I_2 is a coded version of the input test image. Similarly in the anomaly detector 110, the image stored in the nth image store I_n $(1<n<7)$ is a coded version of an input test image. The measurements made on the coded images are collated to form the probability image accumulated in probability image store P_1 or P_{1*} as appropriate. The training procedure described above is designed to optimise the information retained in the coded images. The hierarchical structure of the anomaly detectors 10 and 110 ensures that each coded image deals exclusively with a single length scale in the input image 20.

There are many options to the operation of the anomaly detectors of the invention. The 2-tuples used in the description of the operation of detectors 10 and 110 are simply an example of what may be used. Further optimisation of the detectors 10 and 110 may be achieved by appropriate selection of the 2-tuples to be used for the particular texture being studied.

Alternative embodiments, not illustrated, may be constructed in which a switch is added to each module U intermediate the respective log look-up table L and addition unit A. The switch would enable the contribution from a stage of the detector to pass to the addition unit A or to be prevented from doing so. Thus the contribution from each stage U of the detector may be included in the final "anomaly image", or not, as found best by the operator in order to highlight anomalies in the texture in question. For a more flexible system an additional multiplication unit may be added to each module in this position, and the output from the log look-up table multiplied by some factor G. Such embodiments may be constructed in which the additional multiplication units use like values of the factor G, and this value is variable. Alternatively embodiments may also be constructed in which the factor G is individually variable for each additional multiplication unit. In the former case a single control is provided with which an operator may vary the factor G sent to each additional multiplication unit. In the latter case each additional multiplication unit includes a means by which the operator may alter the factor G used in the respective stage U of the detector.

Further flexibility may be obtained by constructing alternative embodiments, not illustrated, in which the factor F used in the multiplication units X is variable. This would enable the level of the contribution from previous stages of a detector to be determined at each stage. As for the factor G, the factor F may be variable such that each multiplication unit X uses the same value, or such that each multiplication unit X employs a different value. These options would require one or N controls to be provided respectively.

The invention is not limited to the processing of images in 2-tuples. If it is appropriate for the textures being studied images may be processed in the form of combinations of three or more pixels, ie 3-tuples, 4-tuples or higher order tuples. These embodiments may require augmented storage capacities for the image frame stores I and P, mapping look-up tables M, training transputers T, histogram microprocessors H, and log look-up tables L. These capacities must be adequate for the form of pixel data presented. For instance, if processing is to be carried out in 4-tuples, then the form of mapping look-up tables M described for embodiments 10 and 110 is adequate if the image is stored in store I with only 4 bits per pixel. In addition all data buses described in detectors 10 and 100 as being 8, 12 or 16 bit will require replacement with buses of higher capacity. Alternatively serial data buses may be used in place of parallel data buses. If an embodiment of the invention were to use n-tuples (n = 3 or more), then the parameter F used in each multiplication unit X should theoretically be $1/n^2$. If different stages use different orders of tuples, eg 2-tuples in one stage and 3-tuples in another, the theoretical value of F requires calculation. However, as has been said, it is also possible to employ F as a variable parameter which is controllable by an operator.

The form of data transformation used may be selected as appropriate for the texture being processed. It is not limited to a data compression transformation. However the use of a data compression transformation has significant advantages as the stages employed in a detector increase and the order of tuples is increased. The form of the probability transformation may also be selected and is not limited to that described in connection with the detectors 10 and 100.

The number of bits removed from pixel signal values in the truncation units R may also be altered as appropriate for different textures or applications. Alternative embodiments may also be constructed in which truncation units R are omitted, and addresses passing to histogram microprocessors H are not truncated. In order to do this a detector is trained using several images of the same texture. This provides sufficient input to the histogram microprocessors H that the likelihood of a bin contents being zero after histogram accumulation is significantly reduced. An embodiment such as this would require the histogram microprocessors H to have additional memory in order to store the increased number of histogram bins. In addition the data bus from the truncation units R to the histogram microprocessors H would require adequate capacity for the data they were to handle.

There are also many alternatives to the hardware used to implement this invention. For instance the control microprocessors C and C*, and the training transputers T and T* may be replaced by specifically designed electronics. The training transputers T and T* may be omitted and the mapping look-up table provided by a preprogrammed chip trained remotely on the appropriate texture. The detectors 10 and 110 may be constructed with only one control microprocessor in place of the control microprocessors C and C* within each unit U and U*.

Similarly the detectors 10 and 110 may be constructed with only one log look-up table in place of the log look-up tables L and L* in the units U and U*. Further the displays 14 and 14* may be replaced by printers, storage discs or others forms of recording means. It is also possible to employ non-linear functions other than simple logarithms.

Referring now also to Figures 8, 9 and 10, three images tested using a six stage anomaly detector of the invention, and the anomaly images obtained are illustrated. Each image is illustrated as 256x256 pixels with a 256 level grey scale. The anomaly detector uses 2-tuples in each stage as shown in Figure 7. It has switches positioned between each histogram look-up table and the respective log look-up table such that the contribution from each stage may be permitted to contribute to the final anomaly image or not. The anomaly images illustrated in Figures 8, 9 and 10 were all obtained with contributions from the sixth stage only, that is effectively using an 8x8 pixel length scale.

Figure 8a) shows an optical image 120 of a close weave textile. Figure 8b) shows the corresponding anomaly image 122. The anomaly detector employed to obtain the anomaly image 122 was trained using the image 120 before being used to test the same image. As can be seen the closed weave textile has a texture which varies a little over the image 120 as indicated by the mottled appearance of the anomaly image 122. However, there is an anomaly 124 in image 120 where a thread is somewhat displaced from the position that it would be expected to occupy, that is it does not run parallel to the adjacent threads but is displaced to the right. This is clearly highlighted in the anomaly image 122 by the bright spot 126.

Figure 9a) shows an optical image 130 of an open weave textile. Figure 9b) shows the corresponding anomaly image 132. The anomaly detector employed was trained using image 130 before being used to test the same image. The anomaly image 132 indicates a number of anomalies present in the image 130, these are represented as bright spots such as 134. The anomalies may be lengths of thread which are particularly thick or thin, or gaps in the weave that are larger or smaller than the average.

Figure 10a) shows an optical image 140 of a carpet, a square section in the centre of which has been rotated by 180° about its leading diagonal. Figure 10b) shows the corresponding anomaly image 144. The anomaly detector employed was trained using a completely separate image of the carpet (without a section rotated), and was then used to test the image 140. As can be seen from the bright areas, such as 146, in the centre of the anomaly image 144 the anomaly detector clearly identifies the rotated section 142 of the image 140 as being anomalous. Thus, despite the texture of the rotated section 142 being the same as that of the rest of the image 140, the section 142 is identified as being anomalous because of the directional nature of the texture.

CLAIMS

1. An anomaly detector including:

   a) means for receiving input pixel signals,
   b) means for associating pluralities of input pixel signals,
   c) means for performing a probability transformation to replace associated input pixel signals by occurrence probability signals related thereto, and
   d) means for accumulating a probability image comprising image pixel signals arising from functions of occurrence probability signals associated therewith.

2. An anomaly detector according to Claim 1 including:-

   a) means for performing a data transformation on the associated input pixel signals to produce transformed pixel signals,
   b) means for associating pluralities of transformed pixel signals to produce associated transformed pixel signals,
   c) means for performing a probability transformation to replace associated transformed input pixel signals by occurrence probability signals related thereto, and
   d) means for accumulating a probability image comprising image pixel signals arising from combining functions of occurrence probability signals related thereto and derived from input pixel signals with and without the said data transformation.

3. An anomaly detector according to Claim 2 wherein the data transformation is a data compression transformation.

4. An anomaly detector according to Claim 3 wherein the means for performing a data transformation is trainable to alter stored vectors in accordance with a function of pluralities of associated pixel signals and is arranged to implement a data transformation in which associated pixel signals are replaced by respective codes indicating stored vectors associated therewith.

5. An anomaly detector according to Claim 1 including:-

a) transforming means for performing successive cumulative data transformations each involving transformation of respective associated pixel signals produced prior to that transformation and association of pluralities of transformed pixel signals produced by that transformation, the transforming means being arranged to produce successive sets of associated transformed pixel signals,

b) means for performing successive probability transformations to replace associated transformed pixel signals in each set by respective occurrence probability signals related thereto,

c) means for accumulating a probability image comprising image pixel signals arising from combining functions of occurrence probability transformed signals associated therewith and derived from a plurality of the sets of transformed pixel signals.

6. An anomaly detector according to Claim 5 wherein each data transformation is a data compression transformation.

7. An anomaly detector according to Claim 6 wherein the means for performing successive cumulative data transformations is trainable to alter sets of stored vectors in accordance with respective sets of associated pluralities of pixel signals and is arranged to implement data transformations in which successive sets of associated pixel signals are replaced by respective sets of codes indicating stored vectors associated therewith.

8. An anomaly detector according to Claim 5, 6 or 7 of modular construction, and wherein each module includes:-

a) means for receiving input pixel signals,
b) means for associating pluralities of input pixel signals,
c) means for performing a probability transformation to replace associated input pixel signals by occurrence probability signals related thereto,
d) means for performing a data transformation on associated pixel signals to produce transformed pixel signals for output to a succeeding module of like kind to provide input pixel signals therefor,
e) means for accumulating probability image data comprising image pixel signal data arising from functions of occurrence probability signals related thereto and derived from occurrence probability signals produced both in the respective module and a respective succeeding module where available, and
f) means for relaying probability image data to a preceding like module where available or to image storing means otherwise.

9. An anomaly detector according to any one of Claims 5 to 8 wherein the means for accumulating a probability image comprises successive accumulating means each arranged to compute a fraction of the sum of logarithms of functions of all occurrence probability signals arising subsequent to a respective data transformation added to the like of that occurrence probability signal arising immediately prior to that transformation, and to pass the result to a preceding accumulating means where available or to image storing means otherwise, each function being a respective occurrence probability signal which, at least in the case of signals arising subsequent to a first transformation, is divided by a product of summed row and summed column occurrence probabilities associated therewith.

10. An anomaly detector substantially as herein described with reference to Figure 1 or 6.

-35-

Patents Act 1977
Examiner's report to the Comptroller under Section 17 (The Search Report)

Application number
9202752.3

Relevant Technical fields

(i) UK Cl (Edition K) G1A (AAJ)

(ii) Int Cl (Edition 5) G01N 21/88

Search Examiner
J M MCCANN

Databases (see over)
(i) UK Patent Office

(ii) ONLINE DATABASE: WPI

Date of Search
10 APRIL 1992

Documents considered relevant following a search in respect of claims 1-10

Category (see over)	Identity of document and relevant passages	Relevant to claim(s)
X	EP 0243639 A2 (IBM) Column 6 lines 29-50	1
X	EP 0048568 A2 (TRW INC) Page 4 lines 11-21	1